

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-216427

(43)Date of publication of application : 31.07.2003

(51)Int.Cl.

G06F 9/44

G06F 9/45

(21)Application number : 2002-018630

(71)Applicant : HITACHI LTD

(22)Date of filing : 28.01.2002

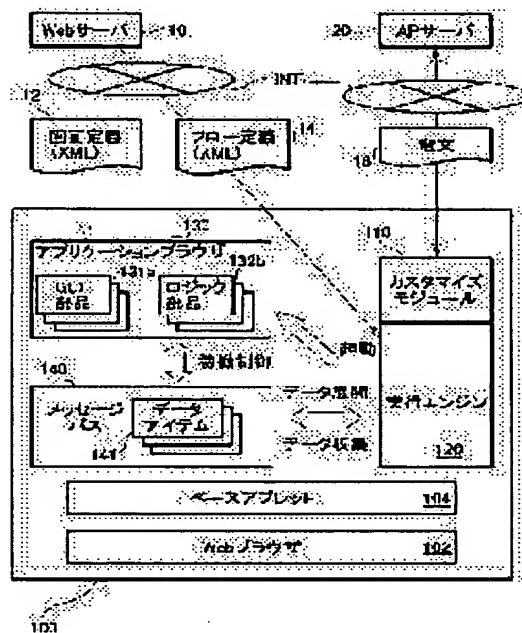
(72)Inventor : MACHIDA ISAMU
MORI YUICHI

(54) CUSTOMIZABLE INFORMATION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To renew and customize processing contents of an information processor without modifying software.

SOLUTION: An execution engine 120 booting on a browser, and an application browser 130 and components 131a and 131b are prepared. The execution engine 120 reads flow definition 14 described in XML (Extensible Markup Language) and executes a command prescribed by a tag in order. When a screen display is demanded, the application browser 130 is called. The application browser 130 reads the screen definition, boots the respective components, and produces a display screen. The software provided with only fundamental and general functions respectively are prepared and the substantial processing contents are given in the XML electronic document, so that the processing contents can be modified without modifying the software.



LEGAL STATUS

[Date of request for examination]

29.06.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2003-216427
(P2003-216427A)

(43)公開日 平成15年7月31日(2003.7.31)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F	9/44	G 0 6 F	9/06 6 2 0 K 5 B 0 7 6
	9/45		9/44 3 2 0 C 5 B 0 8 1

審査請求 未請求 請求項の数33 O L (全 22 頁)

(21)出願番号 特願2002-18630(P2002-18630)

(22)出願日 平成14年1月28日(2002.1.28)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 町田 勇

愛知県尾張旭市晴丘町池上1番地 株式会
社日立製作所情報機器事業部内

(72)発明者 森 有一

神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

(74)代理人 110000028

特許業務法人明成国際特許事務所

Fターム(参考) 5B076 DD06

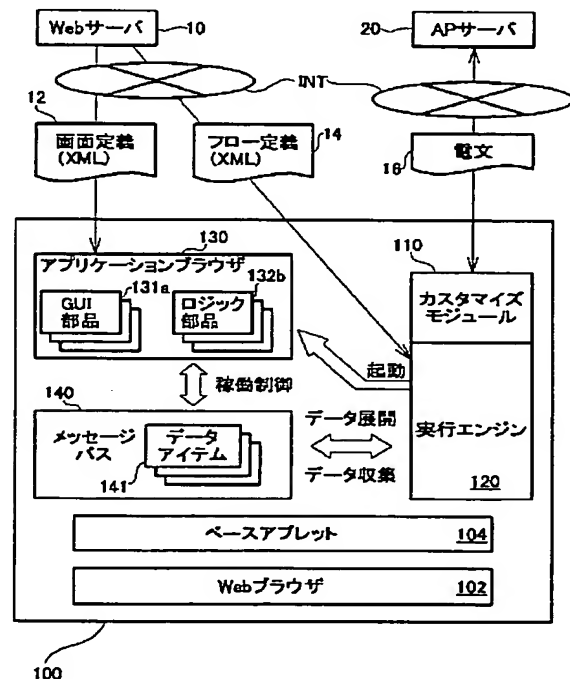
5B081 BB05 DD01

(54)【発明の名称】 カスタマイズ可能な情報処理装置

(57)【要約】

【課題】 ソフトウェアの改変を伴わずに、情報処理装置における処理内容の更新、カスタマイズを実現する。

【解決手段】 ブラウザ上で起動する実行エンジン120、アプリケーションブラウザ130、部品131a、131bを用意する。実行エンジン120は、XMLで記述されたフロー定義14を読み込み、タグで規定されたコマンドを順次実行する。画面表示が要求されている場合には、アプリケーションブラウザ130を呼び出す。アプリケーションブラウザ130は、画面定義を読み込み、各部品を起動して表示画面を生成する。このように、それぞれ基本的、汎用的な機能のみを備えるソフトウェアを用意し、実質的な処理内容をXMLの電子文書で与えることにより、ソフトウェアの改変を伴うことなく、処理内容の改変を実現することができる。



【特許請求の範囲】

【請求項 1】 情報処理装置であって、個別の要素として構成され、予め設定された所定の基本機能を実現する複数種類の基本機能部と、処理に使用すべき基本機能部を特定する形式で、処理内容を規定する電子文書を入力する文書入力部と、該電子文書を解析し、該電子文書に基づいて、逐次、前記基本機能部を稼働させる稼働制御部とを備える情報処理装置。

【請求項 2】 請求項 1 記載の情報処理装置であって、前記基本機能部には、前記情報処理装置に表示されるべき表示画面を生成する画面生成部が含まれ、前記電子文書は、前記表示画面の内容およびその生成タイミングを規定し、前記稼働制御部は、該生成タイミングに応じて前記画面生成部を稼働させる情報処理装置。

【請求項 3】 請求項 1 記載の情報処理装置であって、前記電子文書は、複数の処理内容を規定するとともに、その実施順序を制御するフロー制御情報を含んでおり、前記稼働制御部は、該フロー制御情報に応じた順序で、前記電子文書に規定された基本機能部を稼働させる情報処理装置。

【請求項 4】 請求項 1 記載の情報処理装置であって、前記基本機能部には、前記情報処理装置の表示画面を構成する表示要素を表示するための複数種類の要素表示部が含まれ、前記電子文書は、前記表示画面を構成する各表示要素の種類および位置を規定し、前記稼働制御部は、前記電子文書に基づいて前記要素表示部を稼働させることで前記表示画面を生成する情報処理装置。

【請求項 5】 請求項 1 記載の情報処理装置であって、前記複数の基本機能部には、前記情報処理装置に入力されたデータについて、所定の演算を実行する複数種類の要素演算部が含まれ、前記電子文書は、前記データに施すべき演算の種類を規定し、前記稼働制御部は、前記電子文書に基づいて前記要素演算部を稼働させることで前記データの演算を実行する情報処理装置。

【請求項 6】 情報処理装置であって、個別の要素として構成され、予め設定された所定の基本機能を実現する複数種類の基本機能部と、前記基本機能部の少なくとも一部に対応づけられた所定のオブジェクトを、該対応関係とともに管理するオブジェクト管理部とを備え、前記基本機能部は、前記対応づけられたオブジェクトの状態変化に基づいて起動する機能を有しており、該オブジェクト管理部は、少なくとも一部のオブジェクトに状態変化が生じた場合に、該オブジェクトに関連づ

けられた前記基本機能部に対して、該状態変化を通知する情報処理装置。

【請求項 7】 請求項 6 記載の情報処理装置であって、前記基本機能部には、前記情報処理装置の表示画面を構成する表示要素を表示するための複数種類の要素表示部が含まれ、前記オブジェクト管理部は、複数の表示画面に共通するオブジェクトを管理する共有オブジェクト管理部と、

現に表示されている表示画面に関するオブジェクトを管理する個別オブジェクト管理部とを備える情報処理装置。

【請求項 8】 請求項 6 記載の情報処理装置であって、前記基本機能部は、前記基本機能部とオブジェクトとの対応関係を前記オブジェクト管理部に対し登録可能である情報処理装置。

【請求項 9】 請求項 6 記載の情報処理装置であって、前記オブジェクト管理部は、前記基本機能部の稼働に呼応して、該基本機能部に対応したオブジェクトを生成するオブジェクト生成部と、稼働しているいずれの基本機能部とも対応づけられていないオブジェクトを消去するオブジェクト消去部とを備える情報処理装置。

【請求項 10】 ネットワークに接続され、該ネットワークを介した情報の授受に基づいて所定の処理を行う情報処理装置であって、前記ネットワークを介して送受信される電文に含むべきデータ、および該データのフォーマットを規定する規定文書を記憶する文書記憶部と、前記規定文書を解析し、該規定文書の規定内容に基づいて前記電文の生成または解読を行う電文処理部とを備える情報処理装置。

【請求項 11】 請求項 10 記載の情報処理装置であって、前記所定の処理に関連した複数のデータを統合的に管理するデータ管理部を備え、前記電文処理部は、該情報処理装置から送信される前記電文に含むべきデータを、前記データ管理部から取得する情報処理装置。

【請求項 12】 請求項 10 記載の情報処理装置であって、前記所定の処理に関連した複数のデータを統合的に管理するデータ管理部を備え、前記電文処理部は、前記ネットワークを介して受信した前記電文から抽出したデータを、前記データ管理部に通知する情報処理装置。

【請求項 13】 請求項 10 記載の情報処理装置であって、前記文書記憶部は、複数種類の前記規定文書を記憶しており、

前記ネットワークを介した情報の授受を伴う複数の処理について、各処理で使用するべき前記規定文書を特定した電子文書を入力する文書入力部と、

前記電文処理部は、前記電子文書で特定される規定文書を、現に実行中の処理に応じて、使い分ける情報処理装置。

【請求項 14】 コンピュータによって所定の情報処理を行うためのコンピュータプログラムであって、個別のモジュールとして構成され、予め設定された所定の基本機能を実現する複数種類の基本機能モジュールと、
処理に使用すべき基本機能モジュールを特定する形式で、処理内容を規定する電子文書を入力する文書入力機能と、
該電子文書を解析し、該電子文書に基づいて、逐次、前記基本機能モジュールを稼働させる稼働制御機能とをコンピュータによって実現するためのコンピュータプログラム。

【請求項 15】 請求項 14 記載のコンピュータプログラムであって、
前記基本機能モジュールには、前記コンピュータに表示されるべき表示画面を生成する画面生成モジュールが含まれ、
前記電子文書は、前記表示画面の内容およびその生成タイミングを規定し、
前記稼働制御モジュールは、該生成タイミングに応じて前記画面生成モジュールを稼働させるコンピュータプログラム。

【請求項 16】 請求項 14 記載のコンピュータプログラムであって、
前記電子文書は、複数の処理内容を規定するとともに、その実施順序を制御するフロー制御情報を含んでおり、
前記稼働制御モジュールは、該フロー制御情報に応じた順序で、前記電子文書に規定された基本機能モジュールを稼働させるコンピュータプログラム。

【請求項 17】 請求項 14 記載のコンピュータプログラムであって、
前記基本機能モジュールには、前記コンピュータプログラムの表示画面を構成する表示要素を表示するための複数種類の要素表示モジュールが含まれ、
前記電子文書は、前記表示画面を構成する各表示要素の種類および位置を規定し、
前記稼働制御モジュールは、前記電子文書に基づいて前記要素表示モジュールを稼働させることで前記表示画面を生成するコンピュータプログラム。

【請求項 18】 請求項 14 記載のコンピュータプログラムであって、
前記複数の基本機能モジュールには、前記コンピュータプログラムに入力されたデータについて、所定の演算を実行する複数種類の要素演算モジュールが含まれ、

前記電子文書は、前記データに施すべき演算の種類を規定し、

前記稼働制御モジュールは、前記電子文書に基づいて前記要素演算モジュールを稼働させることで前記データの演算を実行するコンピュータプログラム。

【請求項 19】 コンピュータによって所定の情報処理を行うためのコンピュータプログラムであって、個別のモジュールとして構成され、予め設定された所定の基本機能を実現する複数種類の基本機能モジュールと、

前記基本機能モジュールの少なくとも一部に対応づけられた所定のオブジェクトを、該対応関係とともに管理するオブジェクト管理モジュールとを備え、

前記基本機能モジュールは、前記対応づけられたオブジェクトの状態変化に基づいて起動可能であり、

該オブジェクト管理モジュールは、少なくとも一部のオブジェクトに状態変化が生じた場合に、該オブジェクトに関連づけられた前記基本機能モジュールに対して、該状態変化を通知する機能を有するコンピュータプログラム。

【請求項 20】 請求項 19 記載のコンピュータプログラムであって、

前記基本機能モジュールには、前記コンピュータの表示画面を構成する表示要素を表示するための複数種類の要素表示モジュールが含まれ、

前記オブジェクト管理モジュールは、
複数の表示画面に共通するオブジェクトを管理する共有オブジェクト管理モジュールと、

現に表示されている表示画面に関するオブジェクトを管理する個別オブジェクト管理モジュールとを備えるコンピュータプログラム。

【請求項 21】 請求項 19 記載のコンピュータプログラムであって、

前記基本機能モジュールは、前記基本機能モジュールとオブジェクトとの対応関係を前記オブジェクト管理モジュールに対し登録する機能を備えるコンピュータプログラム。

【請求項 22】 請求項 19 記載のコンピュータプログラムであって、

前記オブジェクト管理モジュールは、
前記基本機能モジュールの稼働に呼応して、該基本機能モジュールに対応したオブジェクトを生成する機能と、
稼働しているいずれの基本機能モジュールとも対応づけられていないオブジェクトを消去する機能とを有するコンピュータプログラム。

【請求項 23】 ネットワークに接続されたコンピュータにおいて、該ネットワークを介した情報の授受に基づいて所定の処理を実現するためのコンピュータプログラムであって、

前記ネットワークを介して送受信される電文に含むべき

データ、および該データのフォーマットを規定する規定文書を管理する文書管理機能と、
前記規定文書を解析し、該規定文書の規定内容に基づいて前記電文の生成または解読を行う電文処理機能とを、コンピュータによって実現するためのコンピュータプログラム。

【請求項 24】 請求項 23 記載のコンピュータプログラムであって、

前記所定の処理に関連した複数のデータを統合的に管理するデータ管理モジュールを備え、

前記電文処理機能は、送信される前記電文に含むべきデータを、前記データ管理モジュールから取得する機能を含むコンピュータプログラム。

【請求項 25】 請求項 23 記載のコンピュータプログラムであって、

前記所定の処理に関連した複数のデータを統合的に管理するデータ管理モジュールを備え、

前記電文処理機能は、前記ネットワークを介して受信した前記電文から抽出したデータを、前記データ管理モジュールに通知する機能を含むコンピュータプログラム。

【請求項 26】 請求項 23 記載のコンピュータプログラムであって、

前記文書管理機能は、複数種類の前記規定文書を管理可能であり、

前記ネットワークを介した情報の授受を伴う複数の処理について、各処理で使用すべき前記規定文書を特定した電子文書を入力する機能と、

前記電文処理機能は、前記電子文書で特定される規定文書を、現に実行中の処理に応じて使い分ける機能を有するコンピュータプログラム。

【請求項 27】 請求項 14～26 いずれか記載のコンピュータプログラムであって、

前記コンピュータにインストールされたブラウザ上で稼働可能に構成されたコンピュータプログラム。

【請求項 28】 コンピュータによって所定の情報処理を実現する情報処理方法であって、

個別のモジュールとして構成され、予め設定された所定の基本機能を実現する複数種類の基本機能モジュールを用意する工程と、

処理に使用すべき基本機能モジュールを特定する形式で、処理内容を規定する電子文書を入力する工程と、

コンピュータによって該電子文書を解析させ、該電子文書に基づいて、逐次、前記基本機能モジュールを稼働させる工程とを備える情報処理方法。

【請求項 29】 コンピュータによって所定の情報処理を実現する情報処理方法であって、

個別のモジュールとして構成され、予め設定された所定の基本機能を実現する複数種類の基本機能モジュールを用意する工程と、

前記基本機能モジュールの少なくとも一部に対応づけら

れた所定のオブジェクトを、該対応関係とともに管理する工程と、

少なくとも一部のオブジェクトに状態変化が生じた場合に、該オブジェクトに関連づけられた前記基本機能モジュールに対して、該状態変化を通知することにより該基本機能モジュールを起動させる工程とを備える情報処理方法。

【請求項 30】 ネットワークに接続されたコンピュータにおいて、該ネットワークを介した情報の授受に基づいて所定の情報処理を実現する情報処理方法であって、

前記ネットワークを介して送受信される電文に含むべきデータ、および該データのフォーマットを規定する規定文書を管理する工程と、

前記規定文書を解析し、該規定文書の規定内容に基づいて前記電文の生成または解読を行う工程とを備える情報処理方法。

【請求項 31】 帳票のイメージデータに基づき所定のイメージ処理を実行するイメージ処理装置であって、

前記イメージ処理に関し、予め設定された所定の基本機能を実現する複数種類の基本機能部と、

使用すべき基本機能部を特定する形式で、前記イメージ処理の処理内容を規定する電子文書を入力する文書入力部と、

該電子文書を解析し、該電子文書に基づいて、逐次、前記基本機能部を稼働させる稼働制御部とを備えるイメージ処理装置。

【請求項 32】 請求項 31 記載のイメージ処理装置であって、

前記基本機能部には、前記イメージデータに基づいて文字データまたは数値データを取得するデータ取得部が含まれ、

前記電子文書は、前記イメージデータから前記取得を行うべき領域、および取得するデータの属性を規定しており、

前記稼働制御部は、該電子文書に基づいて前記データ取得部を稼働することにより、該電子文書で規定されたデータの取得を実現するイメージ処理装置。

【請求項 33】 請求項 31 記載のイメージ処理装置であって、

前記基本機能部には、

前記イメージデータを表示するイメージ表示部、

前記イメージデータに基づいて取得された文字データまたは数値データを表示するデータ表示部、

前記文字データまたは数値データを修正するデータ修正部が含まれ、

前記電子文書は、前記イメージデータと、文字データまたは数値データとの表示態様を規定する表示指示、および外部からの入力に基づき前記文字データまたは数値データの修正方法を規定する修正指示を含み、

前記稼働制御部は、前記表示指示に基づいて前記イメ

ジ表示部およびデータ表示部を稼働することによって前記表示態様を実現するとともに、前記修正指示に基づいて前記データ修正部を稼働することにより前記修正を実現するイメージ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータなどの情報処理装置で行われる情報処理の内容のカスタマイズを実現するための技術に関する。

【0002】

【従来の技術】コンピュータを用いた情報処理は、種々の分野で活用されている。一般に、情報処理の内容は、コンピュータにインストールされるソフトウェアによって決定される。かかる方法では、情報処理の内容を変更するためには、ソフトウェアの更新が必要であり、変更に必要な負担が大きい。

【0003】コンピュータを用いた情報処理の一例として、金融機関で利用される帳票のイメージ処理システムが挙げられる。イメージ処理システムとは、振込用紙など定型的な帳票をスキャンして得られたイメージデータから、いわゆるOCR機能によって氏名、金額などのデータを取得して、取引処理に活用するためのシステムである。イメージ処理システムでは、取得したデータと、イメージデータとを比較して、オペレータがデータの修正を行うこともある。金融機関で使用する振込用紙は、多種多様であり、そこから取得されるデータの処理内容も金融機関によって多種多様である。従って、イメージ処理システムでは、類似しているソフトウェアではあるが、振込用紙の種類や金融機関などに応じた多様なソフトウェアが用いられている。

【0004】こうした処理の多様性に対処するため、いくつかの技術が提案されている。第1の技術は、予めカスタマイズの余地を含んで、ソフトウェアを構築する手法である。振込用紙の種類や処理内容など、予め想定される多様性を包含するように多機能なソフトウェアを構成しておくことにより、実際に使用する機能をパラメータ等で指定するだけで、多様な処理内容を実現することが可能となる。

【0005】第2の技術は、いわゆるオブジェクト指向と呼ばれるプログラミング手法である。かかる手法で構成されたソフトウェアは、比較的シンプルな機能を実行するモジュールの組み合わせで全体の処理を実現しており、各モジュールの置換、追加を比較的容易に行うことができる利点がある。モジュールの置換等により、多様な処理内容を実現することが可能となる。

【0006】第3の技術は、インターネットで多用されているWebプログラミングと呼ばれる技術である。インターネット上のサーバから、クライアントに対し、HTML (Hyper Text Markup Language) のファイルやスクリプト等を送信することにより、クライアントにおい

て種々の処理を実現させることができる。例えば、ブラウザを搭載したクライアントでは、ソフトウェアの変更を伴うだけでなく、HTMLファイルを変更するだけで多種多様な表示画面が実現される。

【0007】

【発明が解決しようとする課題】しかし、情報処理装置において要求されるカスタマイズに十分対応するためには、これらの技術では不十分なことがあった。第1の技術では、ソフトウェアの構築時に予め考慮されている範囲を超えるカスタマイズが要求された場合には、十分に対処することができなかった。第2の技術では、新たな機能を追加する際に、追加機能に対応したモジュールを作成するとともに、追加されたモジュールを呼び出すよう、メインプログラムを改変する必要があり、結局、ソフトウェアの改修が必要とされていた。第3の技術では、HTMLで対応可能な機能は比較的限定されており、十分な情報処理機能を提供するには適していなかった。近年、XML (eXtensible Markup Language) と呼ばれる技術によって、実現可能な機能の拡大が図られてはいるものの、それだけでは、十分な情報処理機能を提供するには至っていない。

【0008】かかる課題は、イメージ処理システムに関わらず、情報処理装置に共通する課題であった。また、情報処理機能の全てをソフトウェア的に実現する装置のみならず、その一部をハードウェア的に提供する装置においても共通の課題であった。ネットワーク等で接続された複数の情報処理装置が連動して情報処理を実現するようなシステムにおいては、アップデートまたはカスタマイズの対象となる装置が多くなるため、上記課題は、特に、深刻なものであった。

【0009】本発明は、これらの課題に鑑みてなされたものであり、ソフトウェアまたは回路の大幅な改変を伴わずに、機能の更新、カスタマイズが可能な情報処理装置を提供することを目的とする。

【0010】

【課題を解決するための手段】上記課題の少なくとも一部を解決するために、本発明では、情報処理装置において、以下に示す構成（以下、「第1の構成」と称する）を適用した。まず、予め設定された所定の基本機能を実現する複数種類の基本機能部を備える。これらの基本機能部は、ソフトウェア的に構成されていてもよいし、電子回路のようにハードウェア的に構成されていてもよいが、それぞれ個別の要素として構成されたものである。個別の要素とは、比較的容易に追加および置換が可能に構成されていることを意味する。

【0011】かかる情報処理装置における処理内容は、電子文書の形式で規定する。電子文書とは、上述した複数種類の基本機能部のうち、処理に使用すべき基本機能部を特定するものである。情報処理装置は、この電子文書を入力し、その内容を解析して処理を実行する。処理

は、電子文書に規定された基本機能部を逐次、稼働させることによって実現される。

【0012】本発明によれば、情報処理装置は、単に、電子文書に従って各基本機能呼び出すという汎用的な機能を備えていることで、種々の処理を実現することができる。基本機能の組み合わせによって実現される実質的な処理内容は、電子文書によって規定されるからである。従って、外部から与える電子文書の内容を変更するだけで、情報処理装置自体のソフトウェアまたはハードウェア構成を大幅に改変するまでなく、情報処理機能の追加、改変、カスタマイズなどを行うことができる。

【0013】本発明の情報処理装置では、基本機能部および電子文書の組み合わせによって、以下に例示する通り、多彩な機能を実現することができる。

【0014】第1の態様として、基本機能部は、表示画面を生成する画面生成部を含んでもよい。このとき、電子文書で、表示画面の内容およびその生成タイミングを規定しておく、情報処理装置は、その生成タイミングに応じて画面生成部を稼働させて画面表示を行うことができる。こうすることで、任意のタイミングで、情報処理装置のインタフェース画面などを表示させることができる。

【0015】第2の態様として、電子文書には、複数の処理内容の実施順序を制御するフロー制御情報を含めてもよい。情報処理装置は、このフロー制御情報に応じた順序で、それぞれの基本機能部を稼働させる。こうすることにより、いわゆる条件分岐、ループなどの複雑なフローを伴う処理を、実現させることができる。フロー制御情報は、情報処理装置が解釈可能であるとともに、電子文書に組み込み可能な情報であれば足りる。

【0016】第3の態様として、基本機能部には、複数種類の要素表示部を含めることができる。要素表示部とは、例えば、線、ボタン、テキストなど表示画面を構成する表示要素を、それぞれ表示するための基本機能部を意味する。このとき、電子文書は、表示画面を構成する各表示要素の種類および位置を規定する。情報処理装置は、この電子文書に基づいて各要素表示部を稼働させると、規定された表示画面を生成することができる。電子文書の内容を変更することにより、多彩な表示画面を比較的容易に生成することが可能である。

【0017】第3の態様は、第1の態様と組み合わせで構成してもよい。つまり、第3の態様における電子文書を読み込んで、各要素表示部を稼働させる機能を、第1の態様における画面生成部が実現するものとしてもよい。こうすることにより、画面生成のタイミングおよび変遷を任意に制御しつつ、多彩な表示画面を実現することができる。

【0018】第4の態様として、基本機能部には、複数種類の要素演算部を含めることができる。要素演算部とは、例えば、四則演算など、情報処理装置に入力された

データについて、所定の演算を実行する基本機能部を意味する。このとき、電子文書は、データに施すべき演算の種類を規定する。情報処理装置は、この電子文書に基づいて各要素演算部を稼働させることで、データの演算を実行することができる。電子文書の内容を変更することにより、多彩な演算を比較的容易に実現することが可能である。

【0019】本発明の情報処理装置では、先に説明した課題の少なくとも一部を解決するための別の構成とし、以下に示す第2の構成を適用してもよい。第2の構成においても、複数の基本機能部を備える点は、第1の構成と共通である。第2の構成では、これらの基本機能部は、各基本機能部に予め対応づけられたオブジェクトの状態変化に基づいて起動する機能を有しているものとする。第2の構成では、情報処理装置は、これらのオブジェクト、および各オブジェクトと基本機能部との関連付けを管理し、オブジェクトに状態変化が生じた場合には、それに関連づけられた基本機能部に対して、状態変化を通知する。

【0020】こうすることにより、各基本機能部の稼働を比較的容易に同期させることができる。また、情報処理装置自体のソフトウェアおよびハードウェアなどに大幅な改変を伴うまでなく、基本機能部を追加、置換等することができる。追加等される基本機能部を正常に動作させるためには、関連づけられるべきオブジェクトを用意すれば足りるからである。

【0021】複数の基本機能部を連鎖的に稼働させるためには、それぞれの基本機能部が動作結果をいずれかのオブジェクトに反映させる機能を有することが好ましい。例えば、第1の機能部、第2の機能部をこの順序で連鎖的に稼働させる場合を考える。第1の機能部は第1のオブジェクトによって稼働し、第2の機能部は第2のオブジェクトによって稼働するものとする。このとき、第1の機能部が、動作結果を第2のオブジェクトに反映する機能を有していれば、その動作に連動して第2の機能部を容易に稼働させることが可能となる。

【0022】第2の構成の情報処理装置において、基本機能部に複数種類の要素表示部が含まれている場合、オブジェクトの管理は、複数の表示画面に共通するオブジェクトと、現に表示されている表示画面に関するオブジェクトとを別個に管理することが好ましい。後者のオブジェクトは、表示画面が切換えられた場合には、管理対象から外すことが可能であるが、前者のオブジェクトは、表示画面が切換えられるか否かにかかわらず保持しておく必要がある。両者を別個に管理することにより、画面の制御を円滑に行うことが可能となる。

【0023】第2の構成では、各基本機能部が、自己とオブジェクトとの対応関係を登録可能とすることが好ましい。こうすることにより、基本機能部の稼働状態、稼働可能性に応じて、オブジェクトとの対応関係が動的に

登録されるため、基本機能部の正常な動作を確保するために要求される対応関係の管理負担を軽減することができる。

【0024】第2の構成の情報処理装置は、各基本機能部の稼働に呼応して、それに対応したオブジェクトを生成することも好ましい。また、稼働中のいずれの基本機能部とも対応づけられていないオブジェクトを消去することも好ましい。つまり、基本機能部の稼働状況に応じて、オブジェクトを動的に生成、消去することが好ましい。こうすることにより、メモリなどの資源を有効活用しながら、オブジェクトを管理することができる。

【0025】本発明の情報処理装置では、先に説明した課題の少なくとも一部を解決するための別の構成として、以下に示す第3の構成を適用してもよい。第3の構成における情報処理装置は、ネットワークに接続され、ネットワークを介した情報の授受に基づいて所定の処理を行うものとする。このとき、情報処理装置は、送受信される電文に含むべきデータ、およびフォーマットを規定する規定文書を記憶しておき、これに基づいて電文の生成または解読を行う。実質的な電文の生成方法は規定文書で定義されるため、情報処理装置は、規定文書に従って電文を生成等する汎用的な機能を有していれば足りる。第3の構成によれば、規定文書の変更により、多彩な電文を比較的容易に生成、解読することが可能となる。

【0026】情報処理装置において、処理に関連した複数のデータが統合的に管理されている場合、送信される電文に含むべきデータをこのデータ管理部から取得したり、受信した電文から抽出したデータをこの管理部に通知したりすることが好ましい。こうすることにより、情報処理装置と外部とのデータの授受を、電文を介して円滑に行うことが可能となる。

【0027】第3の構成において、規定文書は1種類に限定されるものではなく、複数種類を用いても良い。複数種類の規定文書を用いる場合、先に説明した電子文書において、所定で使用すべき規定文書を特定し、規定文書の使い分けを実現してもよい。電子文書と規定文書とは、1対1で対応している必要はなく、例えば、一つの処理において、条件分岐の結果に応じて規定文書を使い分けるものとしてもよい。

【0028】以上で説明した、第1～第3の構成は、それぞれ個別の情報処理装置として適用してもよいし、適宜、組み合わせて適用してもよい。第1の構成および第2の構成は、スタンドアロンの装置に適用してもよいし、ネットワークに接続された装置に適用してもよい。

【0029】本発明の情報処理装置が実施する処理内容は、任意に設定可能であるが、本発明を有効に活用可能な例として、イメージ処理を行うものとしてもよい。つまり、本発明は、帳票のイメージデータに基づき所定のイメージ処理を実行するイメージ処理装置として構成す

ることができる。

【0030】かかる構成では、イメージ処理装置は、イメージ処理に関し、予め設定された所定の基本機能を実現する複数種類の基本機能部を備える。電子文書は、この基本機能部を特定する形式で、イメージ処理の処理内容を規定する。イメージ処理装置は、この電子文書に基づいて、逐次、基本機能部を稼働させることによりイメージ処理を実行する。金融機関で用いられる帳票類は、非常に多彩であり、その処理内容も多様であるため、本発明を適用することにより、イメージ処理装置について金融機関ごとのカスタマイズに要する負担を軽減することができる。

【0031】本発明のイメージ処理装置において、例えば、基本機能部には、イメージデータに基づいて文字データまたは数値データを取得するデータ取得部を含むことができる。この場合、電子文書では、例えば、イメージデータから取得を行うべき領域、および取得するデータの属性を規定し、イメージ処理装置は、電子文書で規定されたデータの取得を実現する。こうすることで、多様な帳票に対して、データの取得を比較的容易に実現することが可能となる。

【0032】本発明のイメージ処理装置において、例えば、基本機能部には、イメージ表示部、データ表示部、データ修正部を含めることができる。イメージ表示部は、イメージデータを表示する基本機能部を意味する。データ表示部は、イメージデータに基づいて取得された文字データまたは数値データを表示する基本機能部を意味する。データ修正部は、文字データまたは数値データを修正する基本機能部を意味する。このとき、電子文書では、例えば、イメージデータ、文字データ等の表示態様、および外部からの入力によって文字データ等を修正する方法を規定し、イメージ処理装置は、これに基づいてイメージ等の表示およびデータの修正を実現するものとしてもよい。本発明を適用することにより、多様な帳票に対して、これらのデータ修正を比較的容易に実現することができる。

【0033】ここでは、イメージ処理装置としての代表的な構成例を示した。先に情報処理装置について説明した種々の構成は、そのままイメージ処理装置に適用可能である。

【0034】本発明の情報処理装置およびイメージ処理装置において、電子文書は、種々の形式を採用することが可能であるが、一例として、XMLで記述された文書（以下、XML文書と呼ぶ）とすることが好ましい。XML文書とは、いわゆるタグを含む、マークアップ言語で記述された文書である。タグは任意に設定可能である。従って、XMLを用いることにより、電子文書に要求される種々のコマンドを、タグの形式で比較的容易に規定することが可能となる。

【0035】本発明は、上述した情報処理装置およびイ

メージ処理装置に限らず、種々の態様で構成することが可能である。例えば、コンピュータによって上述した処理を実現する情報処理方法、またはイメージ処理方法として構成してもよい。これらの処理を実現するためのコンピュータプログラムとして構成してもよい。本発明を、コンピュータプログラムとして構成する場合には、ブラウザ上で稼働可能に構成することが好ましい。例えば、J A V A（登録商標）を用いて構成することが好ましい。こうすることにより、プラットフォームの区別なく稼働可能とすることができるとともに、電子文書としてXML文書を利用しやすいという利点がある。

【0036】更に、これらの種々のコンピュータプログラムを記憶した記憶媒体として構成してもよい。ここで、記憶媒体としては、フレキシブルディスク、CD-ROM、DVD、光磁気ディスク、I Cカード、I・Cチップ、ROMカートリッジ、パンチカード、バーコードなどの符号が印刷された印刷物、コンピュータの内部記憶装置（RAMやROMなどのメモリ）および外部記憶装置など光学的、磁氣的、電氣的にコンピュータが読取り可能な種々の媒体を利用できる。

【0037】

【発明の実施の形態】以下、本発明の実施の形態について、以下の項目に分けて説明する。

A. 第1実施例：

A-1. システム構成：

A-2. フロー定義：

A-3. 画面定義：

A-4. 部品間の連携：

A-5. 電文の授受：

A-6. 実行エンジン：

A-7. アプリケーションブラウザ：

A-8. メッセージバス：

A-10. 起動処理：

A-11. フロー定義のコマンド処理：

A-12. 画面操作処理：

B. 第2実施例（イメージ処理システム）：

B-1. システム構成：

B-2. ソフトウェア構成：

B-3. イメージ処理：

C. 変形例：

【0038】A. 第1実施例：

A-1. システム構成：図1は実施例としての情報処理装置のシステム構成を示す説明図である。情報処理装置100は、汎用のパーソナルコンピュータに図示する各機能ブロックを備えるソフトウェアをインストールすることで構成される。ソフトウェアのインストールのみならず、これらの機能ブロックをハードウェア的に用意することで構成してもよい。

【0039】情報処理装置100は、Webサーバ10、およびAPサーバ20とネットワークINTで接続

されている。ネットワークは、インターネットなど広域的なネットワークとしてもよいし、LAN（Local Area Network）など比較的限定的なネットワークとしてもよい。

【0040】Webサーバ10は、情報処理装置100からのリクエストに応じて、画面定義12、フロー定義14と呼ばれる電子文書を送信する。画面定義12とは、表示画面の内容を規定する電子文書である。フロー定義14とは、画面の遷移など処理の流れを規定する電子文書である。これらの電子文書の内容については、後述する。これらの電子文書は種々の形式を選択可能であるが、本実施例ではXMLで記述されるものとした。

【0041】APサーバ20は、情報処理装置100と連携して種々の処理を実行するサーバである。例えば、端末からの入力データを受けて、ホストコンピュータが種々の処理を実行するシステムの場合、端末が情報処理装置100に相当し、ホストコンピュータがAPサーバ20に相当する。情報処理装置100とAPサーバ20との情報の授受は、電文16によって行われる。電文16の生成方法については後述する。電文16は種々の形式を選択可能であるが、本実施例ではXMLで記述されるものとした。

【0042】なお、本実施例では、情報処理装置100とWebサーバ10、APサーバ20とをネットワークで接続するシステム構成を採用したが、情報処理装置100はスタンドアロンで稼働する装置として構成してもよい。Webサーバ10、APサーバ20は必ずしも必要ではなく、実行すべき情報処理の内容に応じて、情報処理装置100をスタンドアロンで稼働する構成を採用しても良い。

【0043】情報処理装置100に備えられている各機能ブロックの構成について説明する。情報処理装置100には、所定のオペレーティングシステム上で稼働し、HTML等のマークアップ言語で記述されたファイルを閲覧するためのソフトウェア、Webブラウザ102がインストールされている。各機能ブロックは、このWebブラウザ102上で稼働するように構成されている。本実施例では、J A V A（登録商標）を用いることにより、かかる構成を実現した。Webブラウザ102上で稼働可能としたのは、プラットフォーム非依存のソフトウェア構成が可能であるという利点があるからである。Webブラウザ102と無関係に稼働するよう構成しても構わない。

【0044】ベースアプレット104は、Webブラウザ102の上で稼働し、情報処理装置100の各機能ブロック、実行エンジン120、アプリケーションブラウザ130、メッセージバス140などを稼働するためのプラットフォームを提供する。ベースアプレット104およびこれらの各機能ブロックによって情報処理を実現する全体の機構を、本明細書では、XMLアプリケーション

ョンプログラムと称するものとする。本実施例では、これらの機構は、名称の通り、XMLをベースとして構築されているが、XMLアプリケーションプログラムは、広義には、以下で説明する機構の概念を示すものであり、必ずしもXMLベースで構築されているものには限られない。

【0045】ベースアプレット104は、J A V A のアプレットとして構成されている。ベースアプレット104は、その上に構築される各機能ブロックの起動、終了を行う機能、およびコンピュータのキーボード、マウスなどの操作結果（以下、「キーイベント」と称する）を、XMLアプリケーションプログラムの各機能ブロックに受け渡す機能を奏する。

【0046】実行エンジン120は、フロー定義14の内容に基づいて、全体処理を制御する機能を奏する。フロー定義14において、画面の表示、遷移が規定されている場合には、適宜、アプリケーションブラウザ130を起動させる。また、A P サーバ20とのデータ授受が規定されている場合には、電文16のやりとりを行う。電文16の内容は、A P サーバ20によって異なるため、カスタマイズモジュール110によって適応させる。実行エンジン120は、A P サーバ20から受け取ったデータをメッセージバス140に展開したり、逆に、A P サーバ20に送信すべきデータをメッセージバス140から収集したりもする。

【0047】アプリケーションブラウザ130は、実行エンジン120によって起動され、画面定義12に従って、画面を表示し、また、この表示画面を通じた操作を実現する。アプリケーションブラウザ130は、Webブラウザとは別のプログラムである。アプリケーションブラウザ130には、G U I 部品131a、ロジック部品131bと呼ばれる基本的な機能を実現するソフトウェアが用意されている。G U I 部品131aとは、ボタン、線など表示画面を構成する表示要素を生成するためのソフトウェアであり、表示要素の種類に応じて複数用意されている。ロジック部品131bとは、四則演算、入力チェックなど、画面内でのデータに関する処理を実現するためのソフトウェアであり、処理内容に応じて複数用意されている。アプリケーションブラウザ130は、画面定義12に従って、これらの部品を起動させ、表示画面に関する種々の処理を実現する。

【0048】メッセージバス140は、G U I 部品131aおよびロジック部品131bの稼働制御をする。メッセージバス140は、G U I 部品131a、ロジック部品131bに対応づけられたデータアイテム141を複数保持している。各部品は、対応するデータアイテム141の状態変化をトリガとして起動するとともに、その処理結果をデータアイテム141に反映させるよう構成されている。メッセージバス140は、管理下にあるいずれかのデータアイテム141の状態変化が生じた場

合には、それを部品に通知することによって、各部品を起動させる。メッセージバス140は、合わせて処理に必要なデータを保持する機能も奏する。先に説明した通り、実行エンジン120がA P サーバ20との間で授受するデータは、データアイテム141として、メッセージバス140で管理される。

【0049】情報処理装置100は、上述の構成によって、種々の処理を実現する。実行エンジン120、アプリケーションブラウザ130、メッセージバス140は、処理内容に依存したソフトウェアではなく、汎用的な機能を実現するソフトウェアである。処理内容に依存したソフトウェアは、G U I 部品131a、ロジック部品131bなどのモジュールで用意されている。実質的な処理内容は、フロー定義14、画面定義12などの電子文書で、これらの部品を使う順序等を規定する形式で特定される。従って、情報処理装置100は、ソフトウェアの大幅な改変、追加を行うまでなく、電子文書の変更を行うことにより、処理内容を簡便に変更等することができる利点を有している。以下、図中に示した各構成要素について、順次、詳細に説明する。

【0050】A-2. フロー定義：図2はフロー定義14の概要を示す説明図である。フロー定義14とは、実行エンジン120の動作を規定する電子文書であり、本実施例では、XMLによって記述される。

【0051】フロー定義14は、XML宣言文、D T D (Document Type Definition)、および本文から構成される。D T D とは、フロー定義内で使用されるタグを定義する部分である。D T D のみをフロー定義とは、別の文書として用意しても構わない。

【0052】本文は、種々のタグによって、実行エンジン120が実行すべき処理内容を規定している。実行エンジン120は、原則として、フロー定義の本文中のタグを先頭から順に解釈して、タグで規定されるコマンドを実行する。例えば、フロー定義中に、画面生成のコマンドを読み込んだ場合には、アプリケーションブラウザ130を起動することで、画面表示を行う。画面表示には、画面定義12が必要であるため（図1参照）、画面生成コマンドでは、アプリケーションブラウザ130が参照すべき画面定義12のバスも併せて指定される。

【0053】図の下方に、タグによって規定されるコマンド例を示した。コマンドは、処理コマンド、カーソル制御コマンドの2種類に大別される。処理コマンドとは、実行エンジン120が実質的に行う処理内容を規定するコマンドである。例えば、図示する通り、A P サーバ20への送受信、キューへの格納、取りだし、画面の退避、復帰などのコマンドが含まれる。画面の退避、復帰などが、画面生成コマンドに相当する。各コマンドの内容は、実行エンジン120の機能説明と併せて後述する。

【0054】カーソル制御コマンドとは、フロー定義におけるタグの解釈、実行順序を規定するコマンドである。先に説明した通り、実行エンジン120は、原則として先頭から順にタグを実行する。カーソルとは、実行エンジン120が処理を実行しているタグを意味する。カーソル制御コマンドとは、この原則に反して、カーソルの位置を移動させるためのコマンドである。上方の図に示す通り、カーソル移動コマンドを含めることにより、例えば、処理の異動先をタグD、タグEに切り換える条件分岐を実現することができる。

【0055】本実施例では、カーソル制御コマンドとして、図中に示す10種類を用意した。図3はカーソル制御コマンドの概要を示す説明図である。一般にXMLのタグはツリー構造をなしていることから、フロー定義14は、図3に示すようなツリー構造で示すことができる。

【0056】カーソル制御コマンド「FLOWDEF」は、フロー定義のルートを規定するコマンドである。このコマンドに対して、実行エンジン120は、≡main≡という名称のサブルーチンにカーソルを移動する。

【0057】コマンド「SUB」は、サブルーチン宣言のコマンドである。図3の例では、≡main≡、≡yoko≡、≡tate≡という3種類のサブルーチンが用意されている場合を例示した。コマンド「EXITSUB」は、サブルーチン終了のコマンドである。このコマンドに対して、実行エンジン120は、サブルーチンを抜け出し、次のタグにカーソルを進める。

【0058】コマンド「GOSUB」は、サブルーチン呼出のコマンドである。図示する通り、「GOSUB ≡yoko≡」というタグに対して、実行エンジン120は、カーソルをサブルーチン≡yoko≡に移動させる。

【0059】コマンド「SWITCH」は、条件分岐を実現するコマンド、コマンド「CASE」は条件に応じたカーソルの移動先を示すコマンド、コマンド「DEFAULT」はデフォルトの処理内容を示すコマンドである。図3の例では、コマンドSWITCHによって、≡condition≡と称するオブジェクトの状態に応じて条件分岐が行われる。このオブジェクト≡condition≡が≡fine≡の状態にある場合には、CASE≡fine≡以下の処理が実行される。オブジェクト≡condition≡が≡bad≡の状態にある場合には、CASE≡bad≡以下の処理が実行される。オブジェクト≡condition≡が、これらいずれの状態にも該当しない場合には、DEFAULT以下の処理が実行される。

【0060】コマンド「LOOP」は、一定範囲の処理を繰り返し実行するためのコマンドであり、コマンド「EXITLOOP」はループの終端を示すコマンドである。このコマンドに対して、実行エンジン120は、

LOOPからEXITLOOPの間の処理を繰り返し実行する。

【0061】コマンド「NODATA」は、参照したデータの有無に応じて処理内容を切り換えるためのコマンドである。図3の例では、参照したデータが無い場合には、タグA、タグBで規定される処理を実行し、データが存在する場合には、タグCで規定される処理を実行する。

【0062】このように、カーソル制御コマンドをフロー定義中に入れることにより、条件分岐、繰り返し処理などを含む、複雑な処理の記載が可能となる。なお、図2および図3で示した種々のコマンドは例示に過ぎず、コマンドの種類および多少は、これらに限定されるものではない。

【0063】A-3. 画面定義：図4は画面定義の概要を示す説明図である。画面定義12とは、アプリケーションブラウザ130の動作を規定する電子文書であり、本実施例では、XMLで記述される。

【0064】画面定義12の本文は、種々のタグによって、画面の構成や画面で行われる操作内容を規定している。タグは、画面情報、部品情報の2種類に大別される。画面情報とは、画面定義12によって定義される画面に対し、画面IDを与えるタグである。画面定義12は、情報処理装置100で用いられる画面ごとに用意されている。画面IDは、それらの画面を識別するための情報である。

【0065】部品情報は、GUI部品131a、ロジック部品131bを起動するために必要な種々の情報を定義する。GUI部品131aには、文字列表示、文字列入力、画像表示、ボタンなど画面の表示要素を構成する種々のソフトウェアが含まれる。ロジック部品131bには、四則演算、入力条件チェックなど、画面上の操作に関する種々のソフトウェアが含まれる。以下の説明では、混同を回避するため、特に断らない限り、GUI部品131a、ロジック部品131bとして用意されたソフトウェアパッケージを「部品」と称し、これらによって生成された個別の表示要素その他のオブジェクトを、「部品オブジェクト」と称するものとする。

【0066】部品情報は、GUI部品131a、ロジック部品131bを用いて生成される部品に与えられる部品ID、部品のクラス、場所、大きさ、パラメータなどを、それぞれ与えるタグである。画面内には、同じ部品から生成された部品オブジェクトが複数存在する場合がある。例えば、ボタンというGUI部品131aを用いて、ボタンという部品オブジェクトを複数生成させる場合がある。部品IDは、こうして生成された部品オブジェクトそれぞれを個別に管理するための識別情報である。部品の場所、大きさは、例えば、画面上に定義された座標系に基づいて設定することができる。部品のパラメータとしては、例えば、ボタンの形状や表示色、文字

のフォントなどの指定が挙げられる。

【0067】アプリケーションブラウザ130は、画面定義を先頭から順に読み込み、それぞれのタグに従って、処理を実行する。部品1D、部品の位置、大きさなどを指定するそれぞれのタグが見いだされると、アプリケーションブラウザ130は、このタグに従って、各部品を起動するための処理を実行する。この処理によって、画面には、それぞれの部品オブジェクトが表示され、画面上での操作が実現される。なお、図4に示した、タグ、GUI部品131a、ロジック部品131bは、例示に過ぎず、その数および種類は、これらに限定されるものではない。

【0068】A-4. 部品間の連携：図5は部品間の連携を示す説明図である。本実施例では、アプリケーションブラウザ130は、メッセージバス140を介してそれぞれの部品を起動するものとした。

【0069】図示する通り、メッセージバス140は、データアイテムを統合的に管理する。データアイテムは、固有の値を有するとともに、アプリケーションブラウザ130で用いられる部品と関連付けられている。この関連付けは、それぞれデータアイテムごとに登録されている。図中の例では、データアイテムAは、値Aを有するとともに、部品Aと関連づけられている。データアイテムBは、値Bを有するとともに、部品Bと関連づけられている。メッセージバス140は、データアイテムについて、値の変更が生じた場合には、それぞれ関連づけられた部品に、その変更を通知する機能を有している。かかる機能に基づき、本実施例では、各部品を以下に示す手順で起動させることができる。

【0070】一例として、画面定義12において、部品Aの起動が規定されている場合を考える。アプリケーションブラウザ130は、このタグを読み込むと、データアイテムAが保持する値Aを、部品Aの起動トリガとなるべき値に変更する。メッセージバス140は、この変更を、データアイテムAに関連づけられた部品Aに通知する。部品Aは、この通知を受けて起動する。ここで、部品Aは、起動後に、その処理結果に応じてデータアイテムBの値を変更するようプログラミングされているものとする。部品Aの起動によって、データアイテムBの値Bが変更される。メッセージバス140は、この変更を、データアイテムBに関連づけられた部品Bに通知する。部品Bは、この通知を受けて起動する。

【0071】このように、メッセージバス140に管理されたデータアイテムを介して、複数の部品を比較的容易に連動させることができる。例えば、ボタンが押された場合に、ボタンの表示変更と連動して文字列の表示内容を変えることが容易に実現される。

【0072】また、データアイテムを介して各部品を起動させることにより、新たな部品の追加、部品の置換を比較的容易に実現することも可能となる。例えば、部品

Aに連動して起動する新たな部品Cを追加する場合を考える。この場合には、アプリケーションブラウザ130に部品Cのソフトウェアパッケージを用意するとともに、データアイテムBの対象部品に、部品Cを追加すれば足りる。部品Aの起動によってデータアイテムBの値が変更されると、メッセージバス140は、その変更を、部品B、部品Cに通知するため、これらの部品を起動させることができる。このように、部品Aの内容は改変するまでなく、部品Aに連動する部品の追加等を容易に実現することができるのである。

【0073】本実施例では、部品間のこうした柔軟な連携を実現するため、各部品を起動するのに、図5で示した方法を適用した。この利点を考慮しない場合、例えば、部品Aが直接、部品B、部品Cなどを起動する方法を適用しても構わない。

【0074】A-5. 電文の授受：図6は電文の生成、解読方法を示す説明図である。実行エンジン120は、APサーバ20と電文16によってデータを授受する。電文16は、データの種類およびAPサーバ20で用意されるアプリケーションソフトの内容に応じたフォーマットで構成される必要がある。実行エンジン120は、図6に示す構成により、こうした多種多様なフォーマットの電文16を、比較的容易に生成することができる。

【0075】実行エンジン120は、電文16の送受信時には、そのフォーマットを規定するための電子文書、RELAX112を読み込む。RELAX112は種々の形式を採用することができるが、本実施例では、XMLで記述されるものとした。送受信される電文16のフォーマットは単一ではないから、RELAX112も複数用意されている。RELAX112は、予め情報処理装置100内に保持されているものとしてもよいし、Webサーバ10などから供給されるものとしてもよい。

【0076】データの送受信を伴う処理の実行時には、フロー定義14において、使用すべきRELAX112が併せて指定される。図中に一例を示した。ここでは、条件分岐における条件が“fine”に該当する場合には、使用すべきRELAXとしてバスAが規定され、“bad”に該当する場合には、バスBが規定されている。このバス名が、実行エンジン120からカスタマイズモジュール110に受け渡されると、カスタマイズモジュール110がバス名に応じたRELAX112を選択し、それに基づいて送信用の電文16の作成、または受信した電文16の解読を行う。送受信に伴うデータの授受は、メッセージバス140との間で行う。

【0077】実行エンジン120は、それぞれの電文16に依存した生成機能は有さず、RELAX112に基づいた電文16を作成するという汎用的な機能を有している。かかる汎用的な機能を実行エンジン120に備えることにより、実行エンジン120の改変を要せずに、多種多様な電文16を生成することが可能となる。

【0078】A-6. 実行エンジン：図7は実行エンジン120のモジュール構成を示す説明図である。実行エンジン120には、図示する機能ブロックがそれぞれソフトウェア的に構成されている。各機能ブロックは、メイン制御121の制御下で稼働する。

【0079】フロー制御122は、フロー定義を読み込み、その内容に応じて処理を進める機能を奏する。先に図2で説明した通り、フロー定義には、処理コマンド、カーソル制御コマンドが含まれる。フロー制御122は、カーソル制御コマンドの内容を解釈し、これに応じたカーソル移動を実現する。処理コマンドについては、メイン制御121が内容を解釈し、これに応じた機能ブロックをそれぞれ稼働させる。

【0080】コンテキスト制御123は、画面表示などの制御を実現する。「Push」（画面の待避）、「Pop」（画面の復帰）などの処理コマンド（図2参照）に呼応して起動される。コンテキスト制御123は、アプリケーションブラウザ130を適宜、起動させたり停止させたりして、これらの画面表示に関する制御を実行する。

【0081】トランザクション制御125は、APサーバ20との電文の授受に関する通信制御を実行する。

「Upload」（サーバへの送信）、「Transaction」（サーバとの通信）などの処理コマンド（図2参照）に呼応して起動される。本実施例では、電文は、RELAXに基づき生成される。トランザクション制御125は、RELAX選択116、アペンド115、通信プラグイン117を用いて電文の処理を実行する。RELAX選択116は、電文の生成、解説に適用すべきRELAXを選択する機能を奏する。アペンド115は、送信する電文に含めるべきデータについて、ヘッダその他の付加的な情報を生成する。通信プラグイン117は、通信プロトコルに応じて、APサーバ20との電文の送受信を行う。これらの機能は、APサーバ20の機能、通信プロトコルの種類などに応じてカスタマイズが必要であるため、本実施例では、実行エンジン120の汎用的な機能ブロックとは区分されたカスタマイズモジュール110として用意した。

【0082】キュー制御124は、トランザクション制御125、コンテキスト制御123などの同期を採る機能を奏する。キュー制御124は、「Queue」（キューへの格納）、「Dequeue」（キューからの取り出し）などの処理コマンド（図2参照）に応じて起動される。キュー制御124は、トランザクション制御125に受け渡すべき通信スレッド126を生成して、キューに格納する。トランザクション制御125は、適当なタイミングで、このキューに基づいてAPサーバ20との通信を実行する。APサーバ20から受信したデータに応じて画面を表示する場合、通信スレッド126は、結果を保持しておき、適当なタイミングで、画面表

示するためのコンテキスト制御123を起動させる。このようにキューを用いることにより、APサーバ20との通信およびその結果保持を事前に行っておくことが可能となり、画面の遷移等を円滑に進めることができる。

【0083】A-7. アプリケーションブラウザ：図8はアプリケーションブラウザ130のモジュール構成を示す説明図である。アプリケーションブラウザ130には、メイン制御133の制御下で稼働する各機能ブロックが用意されている。

【0084】画面生成部134は、画面定義の内容に基づいて表示画面を構成する部品オブジェクトを生成する機能を奏する。画面表示部135は、メイン制御133からの指示に応じて、生成された部品オブジェクトを順次、表示させ、画面を構成する機能を奏する。画面非表示部136は、逆に、それぞれの部品オブジェクトを停止させて画面を消去する。画面破棄部137は、部品オブジェクト自体を破棄して、画面に関する情報を削除する。バス設定部138は、それぞれの画面を特定するためのバス名、即ち、画面IDを設定する。

【0085】画面で用いられる部品オブジェクトは、部品131によって生成され、オブジェクトテーブル139によって管理される。部品131は、先に説明したGUI部品131a、ロジック部品131bの双方が含まれ得る。アプリケーションブラウザ130で実現される上述の各機能は、オブジェクトテーブル139への各部品オブジェクトの登録、表示、停止などの作用を施すことによって実行される。

【0086】A-8. メッセージバス：図9はメッセージバスのモジュール構成を示す説明図である。メッセージバス140は、データアイテム141を一元的に管理する機能を奏する。本実施例では、メッセージバス140の内部に、複数のバス148を構築し、データアイテム141の種類に応じてバスごとに区分して管理するものとした。これらのバスは、予め固定的に用意されているものに限られず、必要に応じて動的に生成されるものも含まれる。

【0087】バス取得部146はバス148の生成を行う。バスクリア部147は、不要になったバスを消去する。本実施例では、バス148として8種類を用意した。同一種類のバスが複数生成される場合もある。例えば、先読画面に関するバス[6]～[8]は、画面の内容に応じて複数生成される場合がある。

【0088】各バスに対応して、データアイテムを管理するための機能ブロックが設けられる。データアイテム取得部142は、管理対象となっているデータアイテムの状態を示す値を取得する。データアイテム取得部142は、データアイテムを生成する機能も奏する。先に説明した通り、データアイテムは、部品の連携を取るために活用されるオブジェクトであるから、本実施例では、部品の稼働状態に応じてデータアイテムを動的に生成す

るものとした。データアイテム取得部 142 は、部品の稼働等に必要データアイテムが管理対象に含まれているか否かの検索を行い、含まれていない場合には、適宜、データアイテムを生成するとともに、管理対象に加える。こうすることにより、データアイテムの管理に際し、ハードウェア資源を有効活用することができる。

【0089】データ収集部 143 は、実行エンジン 120 が電文に含めるためのデータを、データアイテムから収集する。データクリア部 144 は、データアイテムが保持するデータをクリアする。管理対象となるデータアイテムは、データアイテムテーブル 145 に登録されており、上述の機能は、このデータアイテムテーブル 145 を参照して実行される。

【0090】図中にデータアイテム 141 の機能を列挙した。データアイテム 141 は、図示する 5 つの機能を有している。値設定とは、データアイテムが保持する値を、外部から通知された値に設定し、値の変更が生じた場合には、関連づけられている部品に通知する機能である。先に図 5 で説明した連携を実現するための機能に相当する。値取得とは、外部からの問い合わせに対し、データアイテム 141 が保持している値を応答する機能である。部品登録・管理とは、値の変更が生じた場合に通知すべき部品を管理する機能である。部品登録解除および全部品登録解除は、部品の登録を個別またはまとめて解除する機能である。これらの機能により、データアイテム 141 は、図 5 で説明した通り、部品の起動、および連携を制御することができる。

【0091】データアイテム 141 への部品の登録は、本実施例では、各部品が能動的に行うものとした。図 9 中には部品 131 が有する機能を例示した。

【0092】部品 131 は、図示する 5 つの機能を有している。変更検知とは、データアイテム 141 の値に変更があった場合に、それを検知する機能である。データアイテム 141 からの変更通知を受信する機能に相当する。通知とは、部品オブジェクトの生成、表示、非表示、破棄などを外部に通知する機能である。データアイテム取得とは、部品 131 が関連づけられるべきデータアイテム 141 を検索する機能である。データアイテム監視とは、関連づけられるべきデータアイテム 141 に対し、自己の登録を行う機能である。パラメータ値取得とは、部品オブジェクトの生成時に指定されたパラメータを取得する機能である。

【0093】本実施例では、関連づけられるべきデータアイテム 141 への登録は、部品 131 側で行うものとした。逆に、データアイテム 141 が部品 131 を検索して、登録するものとしてもよい。また、本実施例では、値の変更を、データアイテム 141 が部品に通知するものとしたが、部品がデータアイテム 141 の値を継続的に監視して、その変化を検知するものとしてもよい。

【0094】以上で説明した構成により、情報処理装置 100 は、フロー定義、画面定義、RELAX などの電子文書に従った柔軟な処理、電文生成および解説を実現することができる。また、これらの電子文書の改変により、XML アプリケーションプログラムの基本的な構成の改変を行うまでなく、処理内容を比較的容易かつ柔軟に改変することが可能となる。

【0095】A-10. 起動処理：以下では、XML アプリケーションプログラムを起動して、フロー定義等に応じた処理を実行するまでの各段階での手順を説明する。

【0096】図 10 は起動処理のフローチャートである。この処理は、情報処理装置 100 においてブラウザが起動している状態で、オペレータが XML アプリケーションプログラムベースページの URL を入力することにより、開始される。XML アプリケーションプログラムベースページとは、ベースアプレット 104 を提供する Web ページであり、本実施例では XML で記述される。この URL は、情報処理装置 100 の内部であってもよいし、外部の Web サーバ 10 であってもよい。

【0097】この URL が入力されると、情報処理装置 100 は、ブラウザによって XML アプリケーションプログラムベースページを読み込み（ステップ S10）、そこで提供されるベースアプレットを起動させる（ステップ S11）。ベースアプレットは、図 1 で説明した通り、XML アプリケーションプログラムが稼働するためのプラットフォームを提供するソフトウェアである。

【0098】ベースアプレットの機能により、情報処理装置 100 は、アプリケーションブラウザ 130 を生成する（ステップ S12）。ここで生成されるアプリケーションブラウザ 130 は、XML アプリケーションプログラムが起動するためのプラットフォームとしてのブラウザに相当する。アプリケーションブラウザ 130 は、ベースアプレットで予め指定された初期の画面定義を読み込む（ステップ S13）。この初期の画面定義に基づいて生成される画面を「ベース画面」と称する。アプリケーションブラウザ 130 は、このベース画面上で、実行エンジン 120 を起動させる（ステップ S14）。これにより、XML アプリケーションプログラムが稼働可能な状態が整ったことになる。

【0099】実行エンジン 120 は、フロー定義を読み込み（ステップ S15）、それに基づいてアプリケーションブラウザ 130 を呼び出し、初期画面をオペレータに提示する（ステップ S16）。実行エンジン 120 が読み込むべきフロー定義の URL は、予め設定されている。この URL を、ベースアプレットまたは初期の画面定義で与えるものとしてもよい。

【0100】図 11 は起動後の画面構造を模式的に示す説明図である。XML アプリケーションプログラムは、XML アプリケーションプログラムベースページを読み

込んだWebブラウザ102上で稼働する。XMLアプリケーションプログラムベースページ上では、ベースアプレット104が稼働している。

【0101】ベースアプレット104は、XMLアプリケーションプログラム用のプラットフォームとしてのアプリケーションブラウザ130Aを生成し、XMLアプリケーションプログラムベース画面を提供する。XMLアプリケーションプログラムベース画面上では、実行エンジン120が起動される。通信その他の機能を提供するためのプラグインを必要に応じて起動するものとしてもよい。実行エンジン120は、アプリケーションブラウザ130Bを起動する。アプリケーションブラウザ130Bは、先に説明した通り、種々の部品131を起動して、画面を生成する。部品131には、GUI部品131a、ロジック部品131bの双方が含まれる。アプリケーションブラウザ130Bの起動は、フロー定義によって、画面表示、画面遷移が指示される度に、適宜呼び出されることになる。

【0102】以上の処理によって、XMLアプリケーションプログラムの起動が完了すると、実行エンジン120は、以下に示す通り、フロー定義に基づいて実質的な処理を開始する。

【0103】A-11. フロー定義のコマンド処理：図12は実行エンジンが実行する処理のフローチャートである。実行エンジン120は、フロー定義を読み込み（ステップS20）、そこに含まれるタグの内容を解釈して（ステップS21）、それに応じた処理を実行する（ステップS22）。

【0104】画面表示に関する処理の場合には、実行エンジン120は、コンテキスト制御を実行する（ステップS23）。つまり、アプリケーションブラウザ130を起動して、画面の生成、画面上での操作を実現する。アプリケーションブラウザ130を起動している期間、情報処理装置100の制御は、アプリケーションブラウザ130側に移行する。ステップS23では、実行エンジン120は、アプリケーションブラウザ130が一連の処理を完了するのを待っている。フロー定義で規定された処理が電文処理である場合には、実行エンジン120は、トランザクション制御を実行する（ステップS24）。つまり、RELAXを参照しながら、電文の生成または解釈処理を実行する。図示を省略しているが、その他、実行エンジン120は、タグの内容に応じて、種々の処理を実行可能である。

【0105】実行エンジン120は、以上の処理を、フロー定義に規定された内容が完了するまで（ステップS25）、繰り返し実行する。

【0106】A-12. 画面操作処理：図13は画面操作処理のフローチャートである。実行エンジン120からの呼び出しに応じて、アプリケーションブラウザ130が起動し（ステップS30）、以下に示す処理によ

て、画面表示などの画面操作を実現する。

【0107】アプリケーションブラウザ130は、まず、呼び出し時に指定された画面定義を読み込み（ステップS31）、画面定義の内容に従って、画面表示を行う（ステップS32）。画面表示は、既に説明した通り、画面定義の規定内容に基づいて各部品を呼び出すことによって実現される。

【0108】この状態で、アプリケーションブラウザ130は、キーイベントの入力を待つ（ステップS33）。キーイベントは、情報処理装置100のキーボード、マウスなどの操作結果である。ベースアプレット104が、これらの操作を検知して、その操作内容を表すメッセージを、メッセージバス140に展開する。アプリケーションブラウザ130は、メッセージバス140を監視しているため、この展開結果に基づいてキーイベントを検知することができる。

【0109】キーイベントが完了指示である場合、即ち、画面の遷移、APサーバ20との通信、消去、XMLアプリケーションプログラムの終了などを示している場合、アプリケーションブラウザ130はその動作を終了し、実行エンジン120に制御を戻す（ステップS36）。この際、完了指示が、消去、XMLアプリケーションプログラムの終了など表示画面の消去を要する指示である場合には、アプリケーションブラウザ130は、それまでに表示していた画面を破棄した上で、実行エンジン120に制御を戻す。完了指示が、画面の遷移、通信など、表示画面の消去を要しない指示である場合には、画面の破棄を伴わずに、制御を戻す。

【0110】キーイベントが完了指示でない場合には、アプリケーションブラウザ130は、それぞれのキーイベントに応じてGUI部品、ロジック部品を動作させる（ステップS35）。かかる指示には、例えば、画面上でのデータ入力、ボタン操作、範囲指定などの操作などが含まれる。各部品の動作は、図5で説明した通り、各部品に関連づけられたデータアイテムの値を変更することにより、容易に実現される。

【0111】アプリケーションブラウザ130は、以上の処理によって、画面定義に基づく画面の表示、画面上での操作を実現する。アプリケーションブラウザ130は、実行エンジン120から重複して呼び出される可能性がある。本実施例では、複数のアプリケーションブラウザ130を生成し、それぞれが個別に上述の処理を実行する形で、かかる並列的な処理を実現するものとした。

【0112】以上で説明した第1実施例の情報処理装置100によれば、次に示す種々の利点を得られる。

【0113】第1に、実行エンジン120、アプリケーションブラウザ130が実行すべき実質的な処理内容を、フロー定義14、画面定義12、RELAXなどの電子文書で外部から規定することができる。従って、こ

これらのソフトウェアの改変を行うまでなく、処理内容の変更、拡張を容易に実現することができる。

【0114】第2に、アプリケーションブラウザ130が使用する各部品131は、それぞれ個別のモジュールとして構成されている。従って、新たな部品の追加、部品の置換を比較的容易に実現することができる。

【0115】第3に、各部品は、直接相互に関連している訳ではなく、データアイテムを介して、連携を保っている。このように中間にオブジェクトを介在させて、部品相互の関連付けを行うことにより、各部品の置換、追加を比較的容易に実現することが可能となる。

【0116】第4に、XMLアプリケーションプログラムは、Webブラウザ上で稼働するため、プラットフォーム非依存で、汎用性が高いという利点がある。

【0117】B. 第2実施例 (イメージ処理システム) : 第1実施例では、情報処理システムとして汎用的な構成を例示した。第2実施例では、XMLアプリケーションプログラムの具体的な適用例として、イメージ処理システムについて説明する。

【0118】B-1. システム構成 : 図14はイメージ処理システムの構成を示す説明図である。イメージ処理システムとは、金融機関で利用される振込用紙などの帳票を利用した取引を実行するシステムである。帳票に含まれる情報から得られる文字データまたは数値データ (以下、これらを「帳票データ」と総称する) と、帳票をスキャンして得られたイメージデータとを利用して取引を実行する。

【0119】実施例のイメージ処理システムは、イメージワークフローシステム200と基幹業務システム300の2つを、ネットワークで接続して構築されている。イメージワークフローシステム200は、帳票のイメージデータ、帳票データの登録、補正、紹介を実行する。基幹業務システム300は、取引処理を実行する。基幹業務システム300が実行する取引処理には、帳票を利用しない取引も含まれるが、ここでは説明の便宜上、帳票のデータを利用した取引に絞って説明する。

【0120】基幹業務システム300は、業務サーバ320とホストコンピュータ330を有している。両者を一体的に構成しても構わない。業務サーバ320は、取引に関する業務データを業務データベース310に管理する機能、クライアント250、350などの要求に応じて業務データを供給する機能を奏する。業務データには、例えば、利用者の口座情報、取引履歴などの情報が含まれる。ホストコンピュータ330は、業務データに基づいて、各種取引処理を実現する。

【0121】イメージワークフローシステム200は、イメージサーバ220を有している。イメージサーバ220は、帳票のイメージデータおよび帳票データをそれぞれイメージアーカイブ210として保管し、クライアント250、350などの要求に応じて提示する機能を

奏する。

【0122】イメージデータおよび帳票データは、イメージワークフローシステム200側のクライアント250 (以下、イメージクライアントと称する) によって入力される。イメージクライアント250には、スキャナ204によって、帳票202をスキャンして、イメージデータを取得する。イメージクライアント250は、取得したイメージデータをOCR処理して、帳票データを取得して、イメージサーバ220に登録する。

【0123】登録された帳票データについては、別のオペレータによる確認、補正が行われる。これらの処理は、イメージサーバ220に接続されたいずれかのイメージクライアント250によって行われる。オペレータは、イメージクライアント250で、確認対象となるイメージデータおよび帳票データを並列的に表示させて、両者を比較することで、帳票データの誤りを確認する。誤りがある場合には、適宜、補正してイメージアーカイブ210の内容を更新する。

【0124】B-2. ソフトウェア構成 : 図15はイメージワークフローシステム200のソフトウェア構成を示す説明図である。本実施例では、イメージサーバ220、イメージクライアント250の双方共に、XMLアプリケーションプログラムでそれぞれの処理を実現する。従って、以下に示す通り、それぞれにXMLアプリケーションプログラムを稼働させるための機能ブロックが構築されている。

【0125】XMLアプリケーションプログラムは、イメージサーバ220およびイメージクライアント250のいずれか一方にのみ適用しても構わない。また、基幹業務システム300側にXMLアプリケーションプログラムを適用することも可能である。

【0126】イメージサーバ220には、アプリケーションコア225、アプリケーション部品224が備えられている。アプリケーションコア225は、実行エンジン、アプリケーションブラウザ、メッセージバスの総称である。第1実施例で説明した通り、アプリケーションコア225は、Webブラウザ上で、ベースアプレットによって提供される。

【0127】イメージサーバ220には、アプリケーションコア225が処理に要するアプリケーション部品224および業務フロー定義223も予め用意されている。イメージサーバ220では、画面の表示は行わないため、画面定義は省略しても差し支えない。

【0128】業務フロー定義223は、イメージクライアント250から受け取ったイメージデータおよび帳票データをイメージアーカイブ210に管理するための処理、イメージクライアント250その他のクライアントからの要求に応じて、これらのイメージデータ等を提供するための処理を規定する。

【0129】イメージサーバ220は、Web紹介部2

21 および基幹業務連携部 222 を有している。Web 紹介部 221 は、XML アプリケーションプログラムに非対応のクライアントからの要求に応じて、イメージアーカイブを HTML ファイルの形式で提供する。基幹業務連携部 222 は、基幹業務システム 300 との連携が必要となる処理を実現する。例えば、イメージデータと帳票データの整合だけでは、口座番号が真正なものであるか否かを確認することはできないため、基幹業務連携部 222 は、基幹業務システム 300 に口座番号の真否を確認する。

【0130】イメージクライアント 250 は、同じく、アプリケーションコア 253、アプリケーション部品 252、業務定義 251 を備えている。イメージクライアント 250 では、画面での操作も伴うため、業務定義 251 には、フロー定義、画面定義の双方が含まれる。

【0131】イメージクライアント 250 は、更に、帳票認識部 254 を備えている。本実施例では、帳票認識部 254 は、XML アプリケーションプログラムに依らずに起動するソフトウェアとして用意するものとしたが、XML アプリケーションプログラムを利用してもよい。帳票認識部 254 は、帳票のイメージデータの OCR 処理によって帳票データを取得する。OCR 処理を施す対象となる領域は、帳票ごとに帳票定義 255 によって規定されている。

【0132】イメージクライアント 250 では、帳票データの取得以外の部分を XML アプリケーションプログラムで実現する。フロー定義は、例えば、帳票のイメージデータ取得、帳票認識部 254 によって得られた帳票データとの関連づけ、イメージアーカイブ 210 に格納するためのフォーマット変換などの処理を規定する。画面定義は、これらの処理に必要な操作画面を規定する。

【0133】フロー定義は、帳票データの補正に関する処理も規定する。つまり、イメージアーカイブ 210 からイメージデータ、帳票データをオペレータに提示する処理、帳票データの補正を受け付ける処理などを規定する。画面定義は、補正処理に使用される画面、例えば、イメージデータと帳票データを比較可能な状態で表示する画面、その補正を受け付ける画面などを規定する。

【0134】イメージクライアント 250 とイメージサーバ 220 との通信は、本実施例では、HTTP プロトコルによって行うものとした。両者のデータ授受は、第 1 実施例で説明した電文によって行われる。両者に備えられたアプリケーションコア 253、225 は、それぞれが有する RELAX によって、電文の生成、解読を実行する。

【0135】B-3. イメージ処理：図 16 はイメージ処理全体のフローチャートである。イメージクライアント 250 とイメージサーバ 220 とが連携して実行するワークフロー全体を示した。イメージクライアント 25

0 で実行する処理は、二重枠で示した。

【0136】まず、イメージクライアント 250 が、帳票イメージの読み込みを行い（ステップ S500）、帳票認識を行って帳票データを生成する（ステップ S501）。この帳票データと、イメージデータとを関連づけてデータ登録を行う（ステップ S502）。この処理では、イメージクライアント 250 は、両者を所定のフォーマットに格納して、イメージサーバ 220 に送信する。イメージサーバ 220 は、このデータをイメージアーカイブ 210 の所定領域に格納する。

【0137】データは、種々の形式で格納可能であるが、本実施例では、XML で格納するものとした。つまり、帳票データは、項目ごとに所定のタグとともに XML 形式で記録される。イメージデータは、データを保存しておくパス名が、XML 形式で記録される。こうすることにより、帳票データとイメージデータとを容易に関連づけつつ、一元的に管理することが可能となる。本実施例では、この時点では、補正処理、検証処理が未済というステータスを付してデータを登録するものとした。

【0138】イメージサーバ 220 は、データの登録に続いて、業務連携処理を実行する（ステップ S503）。この処理には、例えば、先に示した通り、口座番号の真否の確認などが含まれる。

【0139】こうしてイメージデータおよび帳票データの登録が完了すると、ワークフローは、これらの補正処理に移行する（ステップ S504）。補正は、イメージクライアント 250 で実行するが、その端末およびオペレータは、データ登録時と異なっても良い。オペレータが、イメージクライアント 250 に「補正処理」を指示すると、イメージクライアント 250 は、イメージアーカイブ 210 に登録されているデータのうち、補正処理、検証処理が共に未済となっているデータを提示する。オペレータは、イメージデータと帳票データとを対比しながら、適宜、帳票データの修正を実行する。こうして修正されたデータが、再び、イメージサーバ 220 に送信されると、イメージサーバ 220 は、イメージアーカイブ 210 の内容を更新する。これとともに、登録時と同様、業務連携処理を実行する（ステップ S505）。この時点では、データのステータスは、検証処理が未済に変更される。

【0140】次に、ワークフローは、イメージデータおよび帳票データの検証処理に移行する（ステップ S506）。処理の手順は、データ補正処理（ステップ S504）と同様である。検証処理は、帳票データの誤りの有無を更に確認するための処理であり、省略することも可能である。

【0141】イメージクライアント 250 が、検証処理が完了したデータをイメージサーバ 220 に送信すると、イメージサーバは、検証処理済みのステータスを付して、イメージアーカイブ 210 の内容を更新すること

によって、このデータ紹介に許可を与える（ステップ S 507）。以上の処理を経て、イメージデータおよび帳票データは、以後、イメージクライアント 250、基幹業務システム 300 からの紹介が可能となる。

【0142】イメージクライアント 250 は、主として XML アプリケーションプログラムによって、上記処理を実現する。実質的な処理内容は、イメージクライアント 250 の業務定義 251 によって与えられることになる。帳票データの生成（ステップ S 501）については、XML アプリケーションプログラムに非依存の帳票認識部 254 を起動する部分、および認識結果を取得する部分で XML アプリケーションプログラムが関連する。

【0143】イメージサーバ 220 も同様に、主として XML アプリケーションプログラムによって、上記処理を実現する。図 16 では、図示を省略したが、実質的な処理内容は、イメージサーバ 220 側の業務フロー定義 223 によって規定される。

【0144】以上で説明した第 2 実施例のイメージ処理システムによれば、帳票データおよびイメージデータを用いた処理に、XML アプリケーションプログラムを適用することができる。金融機関で利用される帳票は、金融機関および取引内容によって、多種多様であり、イメージ処理も多種多様な処理が要求される。XML アプリケーションプログラムは、第 1 実施例で説明した通り、実質的な処理内容を業務定義 251、業務フロー定義 223 で規定することができる。従って、このように多種多様な処理が要求される場合でも、比較的容易に処理内容のカスタマイズ、および改変を実現することができる。

【0145】第 2 実施例では、金融機関のイメージ処理を例にとって説明したが、これに限らず、例えば、保険会社における保険金の支払い業務など、帳票を利用した種々の業務に適用することが可能である。

【0146】C. 変形例：以上の実施例では、いずれも情報処理装置 100、イメージクライアント 250、イメージサーバ 220 などがそれぞれネットワークで接続されている構成を例示した。XML アプリケーションプログラムは、必ずしもネットワークで接続されたコンピュータ上で起動させる必要はなく、スタンドアロンで起動させることもできる。スタンドアロンで起動させる場合には、処理に要するフロー定義、画面定義を予め装置内に記憶しておくものとしてもよいし、CD-ROM などの記録媒体を利用して提供するものとしてもよい。

【0147】以上の実施例では、XML アプリケーションプログラムがソフトウェア的に構築されている場合を例示した。例えば、実行エンジン 120、アプリケーションブラウザ 130、メッセージバス 140 などの機能を実現する回路を用意することによって、ハードウェア的に XML アプリケーションプログラムを実現すること

も可能である。かかる場合でも、回路の大幅な変更、置換を伴わず、処理内容の改変を行うことができる利点がある。

【0148】第 1 実施例で説明した XML アプリケーションプログラムの機能ブロックは、必ずしも全てを備えている必要はない。例えば、実行エンジン 120 の基本的な機能、即ち、フロー定義 14 によって処理を進める機能のみを備え、電文 16 の処理、画面表示に関する処理は、XML アプリケーションプログラムに非依存の方法を採用してもよい。同様に、電文 16 の処理、画面表示に関する処理のみを XML アプリケーションプログラムに依存してもよい。

【0149】以上、本発明の種々の実施例について説明したが、本発明はこれらの実施例に限定されず、その趣旨を逸脱しない範囲で種々の構成を採ることができることはいうまでもない。例えば、以上の制御処理はソフトウェアで実現する他、ハードウェア的に実現するものとしてもよい。

【0150】

【発明の効果】本発明によれば、外部から与える電子文書の内容を変更するだけで、情報処理装置自体のソフトウェアまたはハードウェア構成を大幅に改変するまでなく、情報処理機能の追加、改変、カスタマイズなどを行うことができる。

【図面の簡単な説明】

【図 1】実施例としての情報処理装置のシステム構成を示す説明図である。

【図 2】フロー定義 14 の概要を示す説明図である。

【図 3】カーソル制御コマンドの概要を示す説明図である。

【図 4】画面定義の概要を示す説明図である。

【図 5】部品間の連携を示す説明図である。

【図 6】電文の生成、解読方法を示す説明図である。

【図 7】実行エンジン 120 のモジュール構成を示す説明図である。

【図 8】アプリケーションブラウザ 130 のモジュール構成を示す説明図である。

【図 9】メッセージバスのモジュール構成を示す説明図である。

【図 10】起動処理のフローチャートである。

【図 11】起動後の画面構造を模式的に示す説明図である。

【図 12】実行エンジンが実行する処理のフローチャートである。

【図 13】画面操作処理のフローチャートである。

【図 14】イメージ処理システムの構成を示す説明図である。

【図 15】イメージワークフローシステム 200 のソフトウェア構成を示す説明図である。

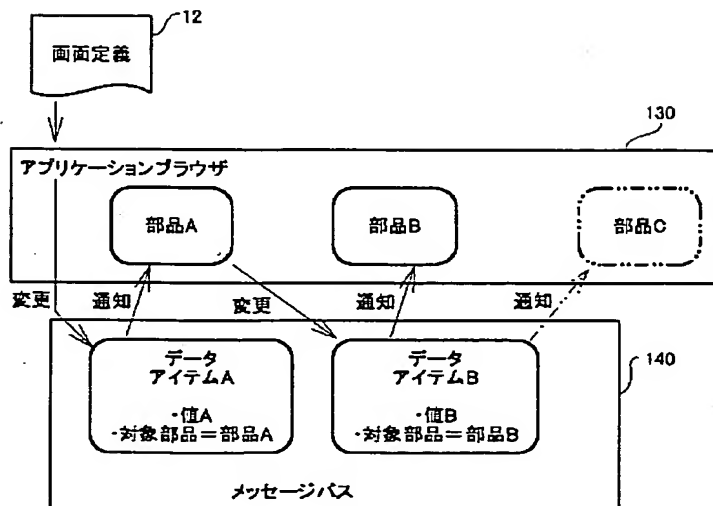
【図 16】イメージ処理全体のフローチャートである。

【符号の説明】

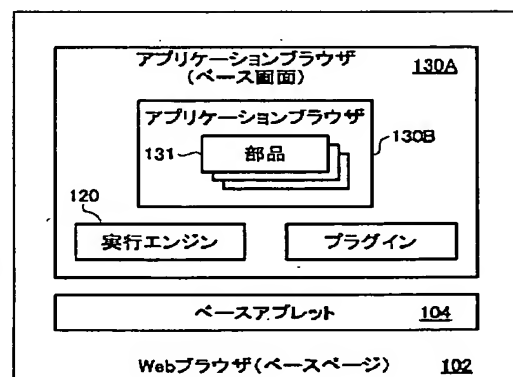
10…Webサーバ
 12…画面定義
 14…フロー定義
 16…電文
 100…情報処理装置
 102…Webブラウザ
 104…ベースアプレット
 110…カスタマイズモジュール
 115…アペンダ
 116…RELAX選択
 117…通信プラグイン
 120…実行エンジン
 121…メイン制御
 122…フロー制御
 123…コンテキスト制御
 124…キュー制御
 125…トランザクション制御
 126…通信スレッド
 130、130A、130B…アプリケーションブラウ
 131…部品
 131a…GUI部品
 131b…ロジック部品
 133…メイン制御
 134…画面生成部
 135…画面表示部
 136…画面非表示部
 137…画面破棄部
 138…パス設定部

139…オブジェクトテーブル
 140…メッセージバス
 141…データアイテム
 142…データアイテム取得部
 143…データ収集部
 144…データクリア部
 145…データアイテムテーブル
 146…パス取得部
 147…パスクリア部
 10 148…パス
 200…イメージワークフローシステム
 202…帳票
 204…スキャナ
 210…イメージアーカイブ
 220…イメージサーバ
 221…Web紹介部
 222…基幹業務連携部
 223…業務フロー定義
 224…アプリケーション部品
 225…アプリケーションコア
 250…イメージクライアント
 251…業務定義
 252…アプリケーション部品
 253…アプリケーションコア
 254…帳票認識部
 255…帳票定義
 300…基幹業務システム
 310…業務データベース
 320…業務サーバ
 30 330…ホストコンピュータ

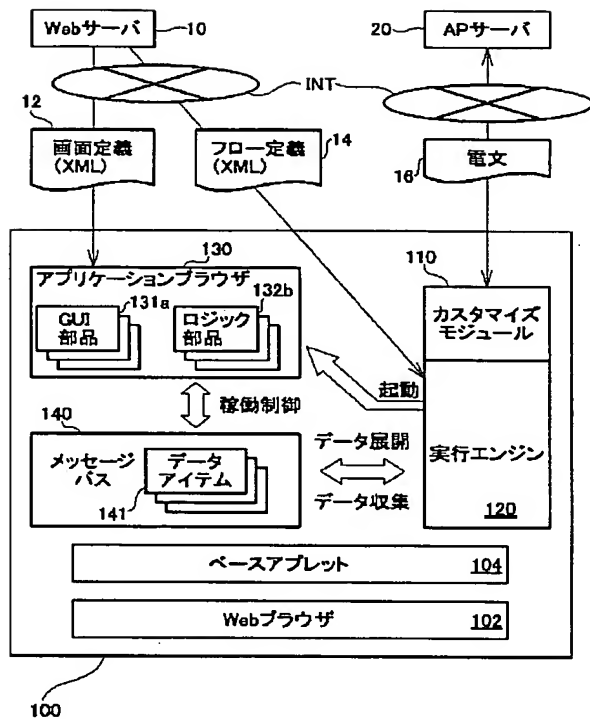
【図5】



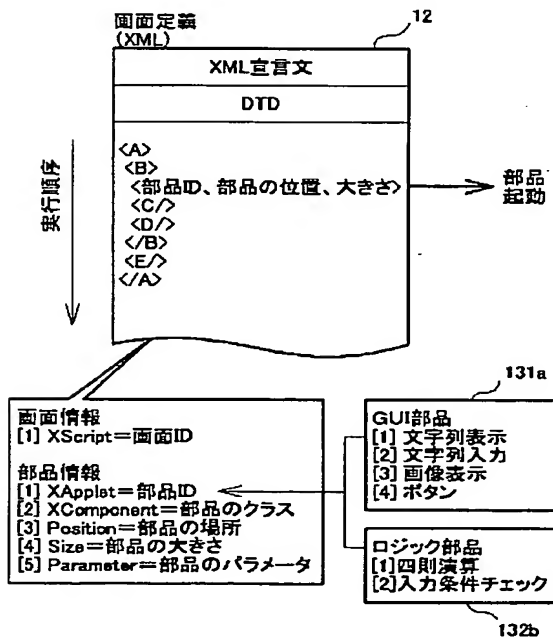
【図11】



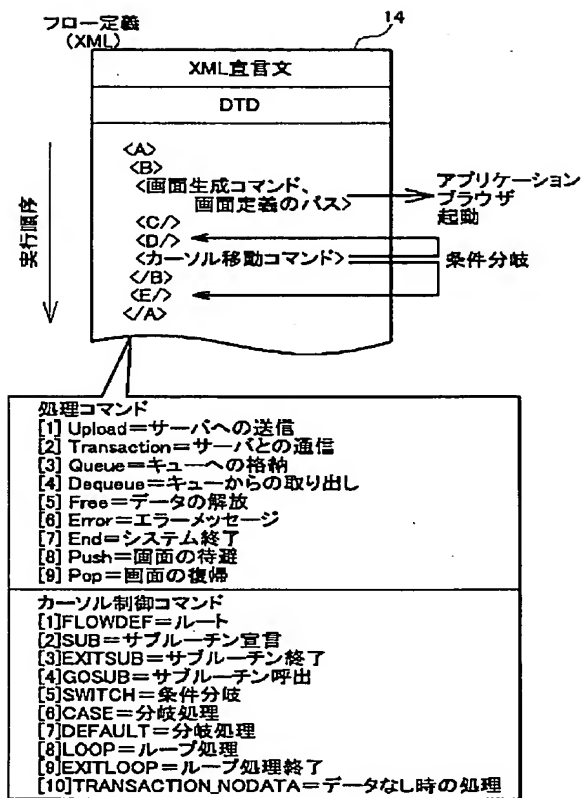
【図 1】



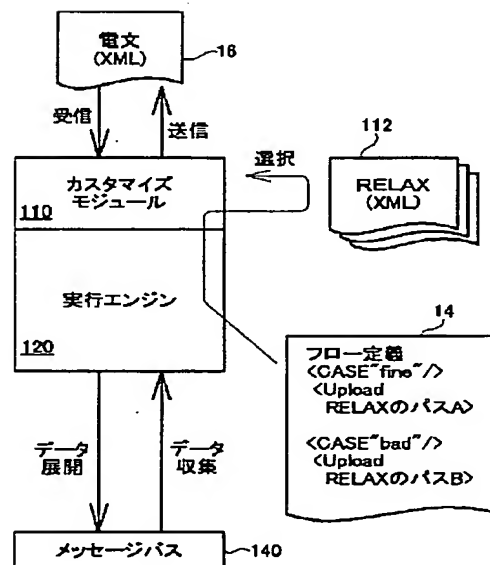
【図 4】



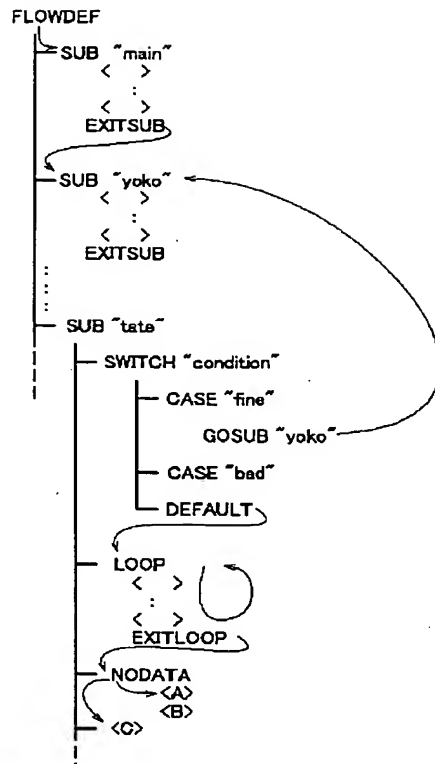
【図 2】



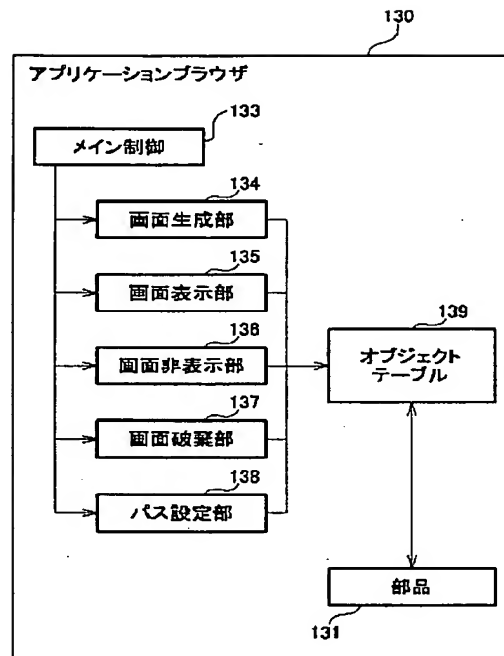
【図 6】



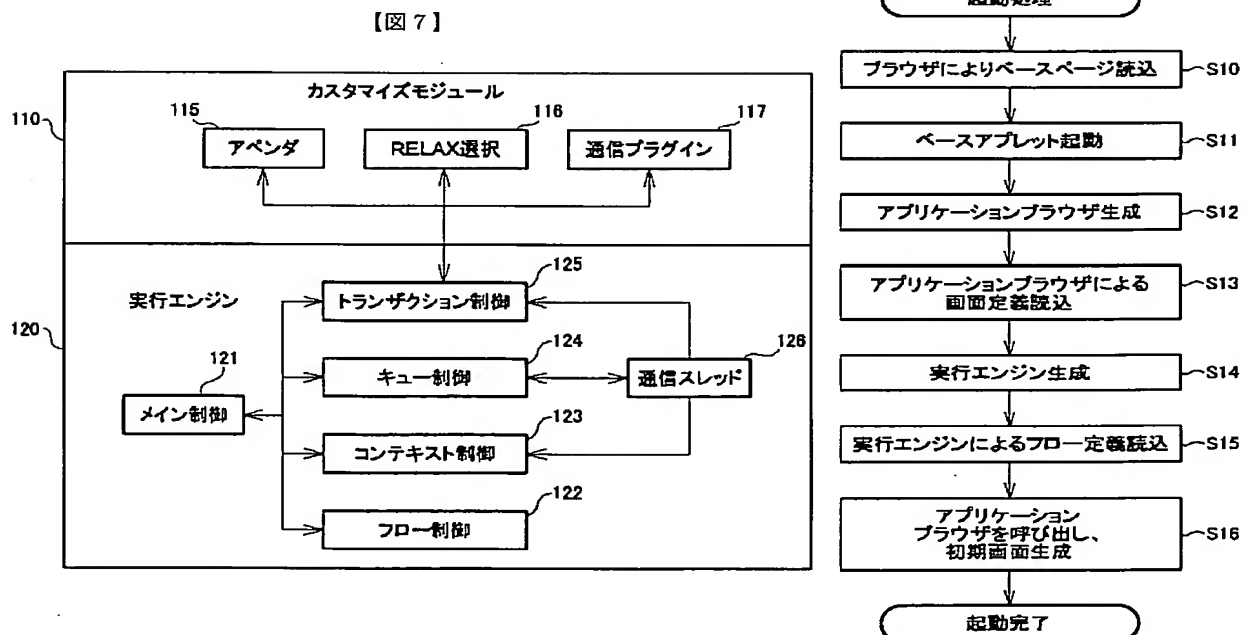
【図 3】



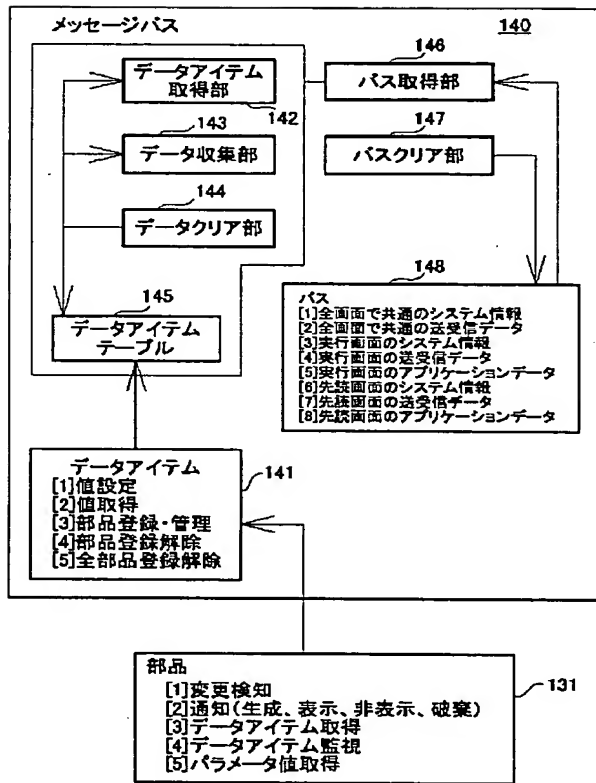
【図 8】



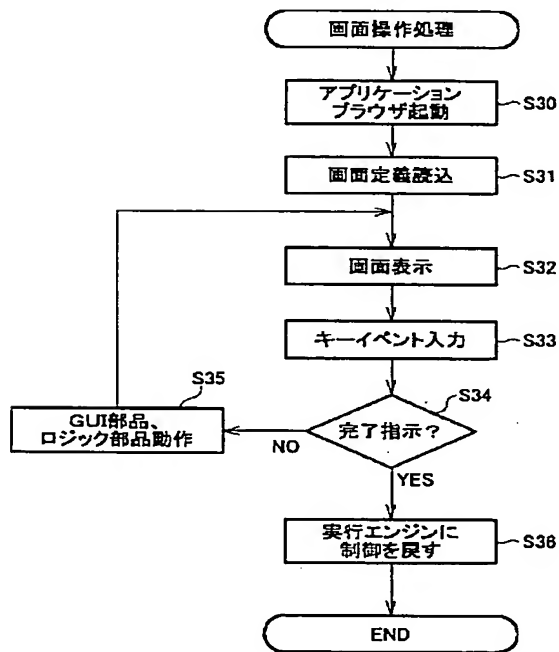
【図 10】



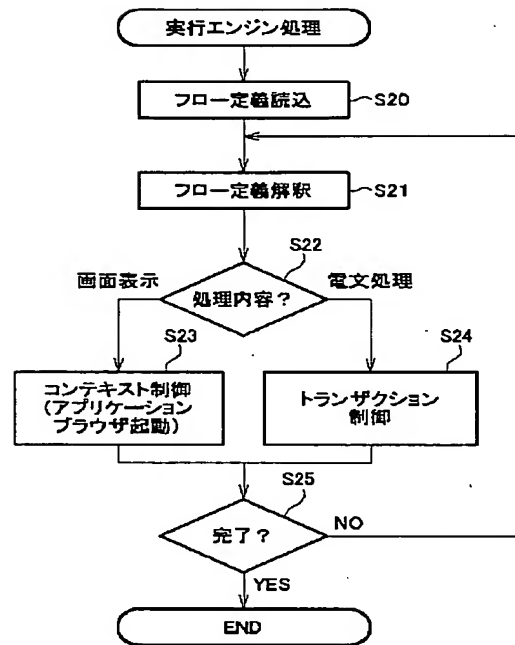
【図9】



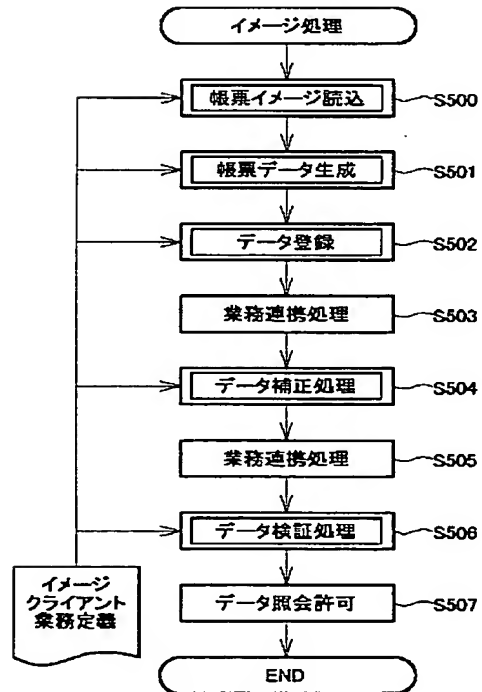
【図13】



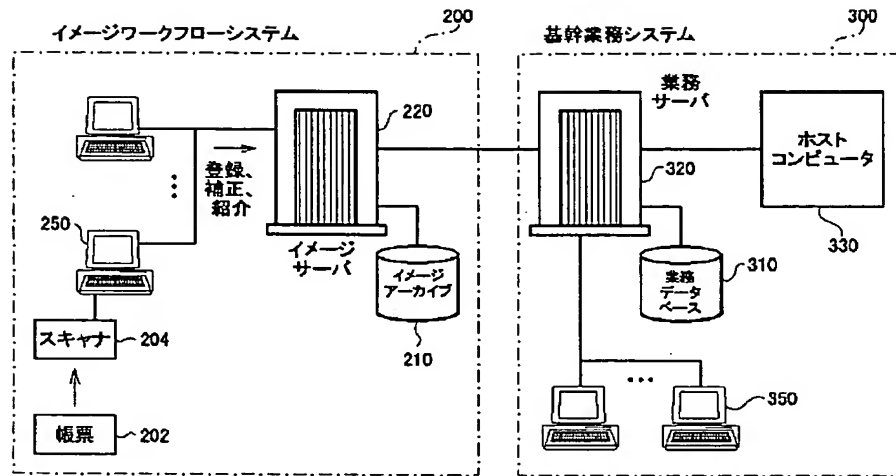
【図12】



【図16】



【図 14】



【図 15】

